Configure Request Debug Logs in Secure Web Appliance

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Request Debug Logs

Configuring the Request Debug Logs

Related Content

Introduction

This document describes the steps to Request Debug Logs in Secure Web Appliance (SWA).

Prerequisites

Requirements

Cisco recommends knowledge of these topics:

- Administrative access to Command Line Interface (CLI) of SWA.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Request Debug Logs

Request Debug Logs in SWA are a specialized log type designed to capture extremely detailed, end to end

debug and up to trace level information for a single, specific HTTP or HTTPS transaction or a client machine. Unlike standard proxy logs that record summarized events across many requests, Request Debug Logs aggregate debug output from all Web Proxy modules involved in processing a particular request (such as authentication, URL filtering, decryption, malware scanning, and reputation services) into one correlated log stream. This log type is intended purely for deep diagnostics and can only be created via the CLI, not through the GUI.

Request Debug Logs are essential when troubleshooting complex or intermittent proxy issues where standard logs lack sufficient detail. They allow administrators and Cisco TAC to trace exactly how a single request was handled at every processing stage, making it possible to pinpoint root causes such as unexpected policy matches, scanning delays, authentication failures, or inconsistent verdicts between engines. Because the log focuses on one transaction, it provides maximum visibility without the operational overhead and performance impact of enabling debug logging across all proxy modules system-wide. This makes Request Debug Logs a precise, efficient, and low risk, diagnostic tool during advanced investigations.

Configuring the Request Debug Logs

Step 1. Log in to CLI, run **logconfig** and choose **new**.

Step 2. Select the number associated with **Request Debug Logs** and press **Enter**.

Step 3. Enter the name for the log.

Step 4. Choose **Trace** as the logging level.

Step 5. Choose modules where requested to collect the enhanced logging. Multiple selections can be made in the form of a comma separated or range list (such as 1,3,4 or 3-7).



Tip: If no specific module is requested by the TAC, it is best to select all modules. (For example: 1-30).

Step 6. Specify the number of requests for which enhanced logging is to be enabled. Once this number of requests has been captured, logging automatically stops.



Note: It is important to select a reasonable value based on the traffic conditions during troubleshooting. For example, if a dedicated test machine is being used and background traffic is minimal, a lower number of requests is sufficient. However, in environments with higher background activity (such as operating system updates, browser background requests, or applications like Webex), choosing a higher value ensures that the relevant transaction is captured.

Step 7. Define the request matching criteria for enhanced logging by selecting either the **Client IP address**, **Destination IP address**, or **Destination domain**.

Note: In most cases, it is recommended to select the **Client IP address,** even when troubleshooting access to a single website. This approach ensures that all web requests generated during page load are captured, including background requests to additional URLs that possibly are not immediately visible. However, this method is most effective when using a dedicated test machine with minimal background internet traffic. In environments where the client generates significant additional traffic (such as operating system updates, browser background services, or applications like Webex) it is better to filter by **Destination domain** or **Destination IP address**.

Tip: If the exact point of failure is unknown, browser **HAR** logs can be collected to identify the specific URL or domain exhibiting issues (for example, page load failures or high latency), and that domain can then be configured in the Request Debug Log criteria.

Step 8. Choose the method to retrieve the logs. If you select **FTP Poll** the logs stores on the SWA.

Step 9. Define the File name to use for log files, or press Enter to accept the current generated file name.

Step 10. Select No for the time-based log files rollover, since the logging stops after the defined number of requests were met.

Step 11. Define the maximum file size in Bytes, or press Enter to accept the current value.

Tip: Defining a larger log file size can make logs more difficult to download and review. Instead of increasing the size of individual log files, it is recommended to increase the number of log files (Next Step). This approach improves manageability while ensuring that all required debug information is captured without creating overly large files.

Step 12. Configure the maximum number of log files based on the number of proxy modules selected for logging in **Step 5** and the **request matching criteria** defined in **Step 7**. Selecting a reasonable file limit is important to ensure that all relevant debug information is captured without prematurely stopping logging, which could result in incomplete or missing logs.

Step 13. Select **No** when prompted with **Should an alert be sent when files are removed due to the maximum number of files allowed?** This prevents unnecessary alerts during normal log rotation, especially when Request Debug Logs are generated intentionally for troubleshooting purposes.

Step 14. Select **No** when prompted with **Do you want to compress logs (yes/no)?** This keeps the log files uncompressed, making them easier to review and analyze during troubleshooting.

Step 15. Press Enter to exit the wizard.

Step 16. Type **commit** and press **Enter** to save the changes.

```
SWA_CLI> logconfig

Currently configured logs:
1. "accesslogs" Type: "Access Logs" Retrieval: FTP Poll
2. "adc_logs" Type: "ADC Engine Logs" Retrieval: FTP Poll
...
[Output removed to simplify readability]
...
55. "welcomeack_logs" Type: "Welcome Page Acknowledgement Logs" Retrieval: FTP Poll

Choose the operation you want to perform:
- NEW - Create a new log.
- EDIT - Modify a log subscription.
- DELETE - Remove a log subscription.
- HOSTKEYCONFIG - Configure SSH host keys.
- AUDITLOGCONFIG - Adjust settings for audit logging.
[]> new

Choose the log file type for this subscription:
1. ADC Engine Framework Logs
2. ADC Engine Logs
...
[Output removed to simplify readability]
...
53. Request Debug Logs
...
[Output removed to simplify readability]
...
[1]> 53

Please enter the name for the log:
[]> Request_Debug_Logs

Log level:
1. Critical
2. Warning
3. Information
4. Debug
5. Trace
[3]> 5

Choose modules where enhanced request logging is to be performed.
Multiple selections can be made in the form of a comma separated or range list (e.g. 1,3,4 or 3-7)
Choosing the Default Proxy will enable enhanced logging across modules:
1. Default Proxy
2. Access Control Engine
3. Proxy Configuration
4. Disk Manager
5. Memory Manager
6. McAfee Integration Framework
7. Sophos Integration Framework
8. Webroot Integration Framework
9. Webcat Integration Framework
10. Connection Management
11. Authentication Framework
12. HTTPS
13. FTP proxy
14. WCCP Module
15. License Module
16. SNMP Module
17. WBRS Integration Framework
18. Logging Framework
```

```
19. Data Security Module
20. Miscellaneous Proxy Modules
21. DCA Engine Framework
22. AVC Engine Framework
23. Cloud Connector
24. SOCKS Proxy
25. Advanced Malware Protection
26. ArchiveScan module in proxy
27. Web Traffic Tap module in proxy
28. Bandwidth Control
29. Http2 proxy
30. ADC Engine Framework
[1]> 1-30

Please enter the number of requests for which to perform enhanced logging:
[1]> 100

Choose the request criteria for logging:
1. Client IP Address
2. Destination Domain
3. Destination IP Address
[1]> 1

Specify source IP address
[]> 10.20.3.15

Choose the method to retrieve the logs:
1. FTP Poll
2. FTP Push
3. SCP Push
[1]> 1

Filename to use for log files:
[Request_Debug_Logs.text]>

Do you want to configure time-based log files rollover? [N]>

Please enter the maximum file size:
[10485760]>

Please enter the maximum number of files:
[10]> 50

Should an alert be sent when files are removed due to the maximum number of files allowed? [N]>

Do you want to compress logs (yes/no)
[n]>

Currently configured logs:
1. "Request_Debug_Logs" Type: "Request Debug Logs" Retrieval: FTP Poll
2. "accesslogs" Type: "Access Logs" Retrieval: FTP Poll
3. "adc_logs" Type: "ADC Engine Logs" Retrieval: FTP Poll
...
[Output removed to simplify readability]
...
56. "welcomeack_logs" Type: "Welcome Page Acknowledgement Logs" Retrieval: FTP Poll


SWA_LIC> commit

Warning: In order to process these changes, the proxy
process will restart after Commit. This will cause a brief
```

```
interruption in service. Additionally, the authentication
cache will be cleared, which might require some users to
authenticate again.
```

Related Content

- User Guide for AsyncOS 15.2 for Cisco Secure Web Appliance
- Use Secure Web Appliance Best Practices
- Access Secure Web Appliance Logs
- Configure SCP Push Logs in SWA with Microsoft Server